AMENDMENTS TO THE SPECIFICATION

On page 3, please replace the paragraph starting at line 9 with the following paragraph:

Referring to FIG. 1, an example of a simple statechart 2 is shown using Stateflow®. ~~Attached in Appendix I are portions of the user manual for Stateflow®, included herein by reference, containing descriptions and definitions of a number of terms use herein.~~ States 4, 6, 8 and 10 are shown with transitions 12,14,16,18 20 and 22 modeling the states of an automobile transmission. The following section provides descriptions and definitions of a number of terms used in this application.

On page 3, please insert the following paragraphs at line 14:

A state diagram is a graphical representation of a finite state machine where states and transitions form the basic building blocks of the system.

A state describes a mode of an event-driven system. The activity or inactivity of the states dynamically changes based on events and conditions. Each state has hierarchy. Each state may have a parent state and/or a child state. Each state has a higher hierarchy than its child state but a lower hierarchy than its parent state.

A transition is a graphical object that can link one object to another. One end of a transition is attached to a source object and the other end to a destination object. The source is where the transition begins and the destination is where the transition ends.

A connective junction is a decision point in the system. A connective junction provides an alternative way to represent desired system behavior. A connective junction is a graphical object that simplifies a state diagram representation and facilitates generation of efficient code.

A history junction provides means to specify the destination substate of a transition based on historical information. If a superstate has a history junction, the transition to the destination substate is defined to be the sutbstate that was most recently visited. The history junction applies to the level of the hierarchy in which it appears.

A data object/item can store numerical values for reference in a state diagram. Data objects/items are nongraphical objects and are not represented in the figure of the state diagram.

A condition is a Boolean expression specifying that a transition occurs given that the specified expression is true.

A graphical function is a function defined by a flow graph. Graphical functions are similar to textual functions, such as MATLAB and C functions. Like textual functions, graphical functions can accept arguments and return results. Unlike MATLAB and C functions, graphical functions are objects that reside with the state diagram that invokes them. Graphical functions are easier to create, access, and manage than textual functions, whose creation requires external tools and whose definition resides separately from the state diagram.

On page 3, please replace the paragraph starting at line 22 with the following paragraph:

The function prototype 34 specifies the syntax for invoking the function in state and transition actions. In an exemplary implementation described below, it has a function name 38, a parameter list 40 listing arguments passed to the function when it is invoked, and a return parameter 44 representing a list of values returned by the function. Other structures may be used to accomplish a similar result. The number of parameters passed to the function may be any number. <u>The number of output parameters returned by the function may be any number as well.</u> In the described embodiment, actions that invoke a graphical function pass arguments 40 and 42 to the function in the same order that they appear in the function's parameter list; however, other argument passing schemes could also be used beneficially.

On page 4, please replace the paragraph starting at line 29 with the following paragraph:

(i) A statechart can export its functions. The functions exported by a chart can be invoked anywhere in the state machine in which the chart appears, including other charts ~~that~~ define<u>d in</u> the state machine.

On page 5, please replace the paragraph starting at line 1 with the following paragraph:

(ii) A graphical function shadows any functions of the same name defined in ancestors of that graphical function's parent state or chart. In other words, a state or transition that invokes function A will get the version of A defined closest to it in the state diagram hierarchy. For example, FIG. 9 shows a transition condition 134 in state C 130 that invokes a graphical function name f1. <u>The transition condition 134 is a condition of transition 138 to make a transition from state D 140 to state E 144.</u> The chart contains two definitions of f1, one 124 defined in state B 126, the other 120 defined in state A 12<u>8</u>~~6~~. In this example, state B's definition of f1 is the definition that is invoked when transition condition 134 is evaluated in state C <u>130</u>. This is because state B <u>126</u> is a more immediate ancestor of state C <u>130</u> than is state A <u>128</u>.

On page 5, please replace the paragraph starting at line 9 with the following paragraph:

A state or transition action may invoke a graphical function by replacing the formal parameters of the prototype with actual arguments and assigning the result to a variable. For example, FIG. 4 shows a defined graphical function 60 named f1 62 that multiplies its arguments 64 and 66 <u>according to expression 69</u> and an invocation (call) 76 of f1 62 in the entry action 70 of a state 72 named A 74. Note that the return parameter 68 in the function prototype of f1 62 need not have the same name as the return parameter in the invocation of the function 78.

On page 8, please replace the paragraph starting at line 12 with the following paragraph:

The term scope refers to the role (argument or return value) of the data items specified by the function's prototype. <u>The term scope can also refer to a data item's visibility.</u> In this sense, arguments and return values have local scope. They are visible only in the flow diagram that implements the function.